Table of Contents


```
%-------------------------------------------
% AER 715 Avionics and Systems
% Lab 4 – Flight Control - Control System Design
% Sharvani Yadav: 501108658
% Daniel Mielnik: 501118927
%------------------------------------------- %
```

Lab Intorduction

In the previous lab, a mathematical model for the vertical (elevation) dynamics of a 3-DOF helicopter was developed. This lab extends that work by focusing on the horizontal travel dynamics. In the first part of the lab, an existing Simulink model was modified to control the helicopter's travel, testing the model's performance by validating a PID controller using various blocks from the Simulink library. In the second part, a PID controller was designed in SISOTOOL for the system's second-order dynamics, with target performance criteria: a settling time of no more than 12 seconds, an overshoot of less than 5%, and zero steady-state error.

Lab Exercises –

```
clc;
```

Constants for Heli 4

```
Mh = 1.450; % Mass of Heli Body (kg)
Mc = 1.918; % Mass of CW (kg)
La = 25.75/39.37; % Distance from Pivot to Helecopter body center (m)
Lb = 18.5/39.37; % Distance from Pivot to conterweight center (m)
Lh = 6.933; % Distance from pitch axis to rotor center (m)
Kf = 0.140; % Motor-Prop Force Constant (N/V)
Krt = 0.0027; % Motor-Prop Torque Constant (Nm/V)
epsilon = -26:1:30; % Elevation Range (Deg)
epsilon_0 = -25.75; % Elevation Start (Deg)
lambda = 0:1:360; % Travel Range (Deg)
g = 9.81; % Gravity constant (m/s^2)
Wh = Mh*g; % Weight of Heli Body (N)
Wc = Mc*g; % Weight of CW (N)
```

```
Je = (Mh * La^2) + (Mc * Lb^2) % Elevation Axis (kg-m^2)

% Open loop transfer function
G4_elev1 = tf(La*Kf, [Je, 0, 0])
G4_elev2 = tf(0.0294, [1.0000, 0.1538, 1.3288])
G4_elev3 = tf(0.0911, [1.0000, 3.1682, 1.7900, 4.0932])
G4_trav = tf(0.002362, [1 0 0])

% Closed loop transfer function
T = tf(La*Kf, [Je, 0, 0] + La*Kf)

% Load Sisotool data
% sisotool('SisoElev1.mat') % originaly sisotool('rlocus', G4_elev1)
% sisotool('SisoElev2.mat') % originaly sisotool('rlocus', G4_elev2)
% sisotool('SisoElev3.mat') % originaly sisotool('rlocus', G4_elev3)
% sisotool('Trav.mat') % originaly sisotool('rlocus', G4_trav)


Je =

    1.0438


G4_elev1 =

   0.09157
  ---------
  1.044 s^2

Continuous-time transfer function.

G4_elev2 =

          0.0294
  ----------------------
  s^2 + 0.1538 s + 1.329

Continuous-time transfer function.

G4_elev3 =

               0.0911
  --------------------------------
  s^3 + 3.168 s^2 + 1.79 s + 4.093

Continuous-time transfer function.

G4_trav =

  0.002362
  --------
    s^2

Continuous-time transfer function.
```

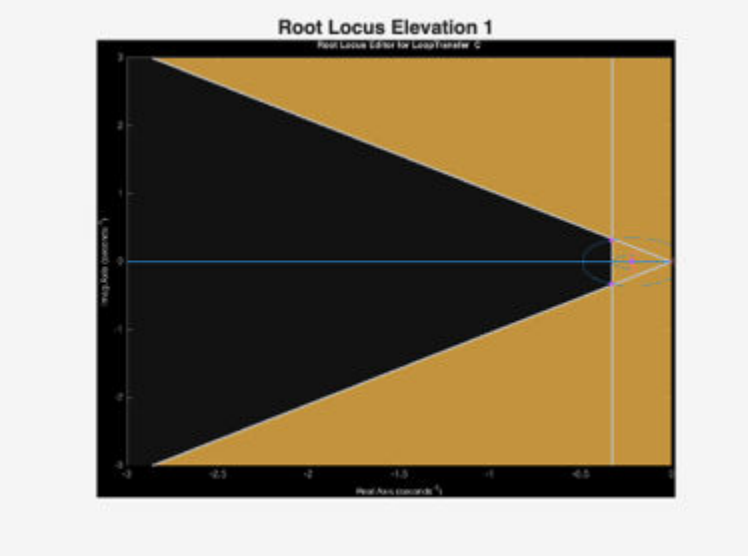
```
T =

              0.09157
  -------------------------------
  1.135 s^2 + 0.09157 s + 0.09157

Continuous-time transfer function.
```

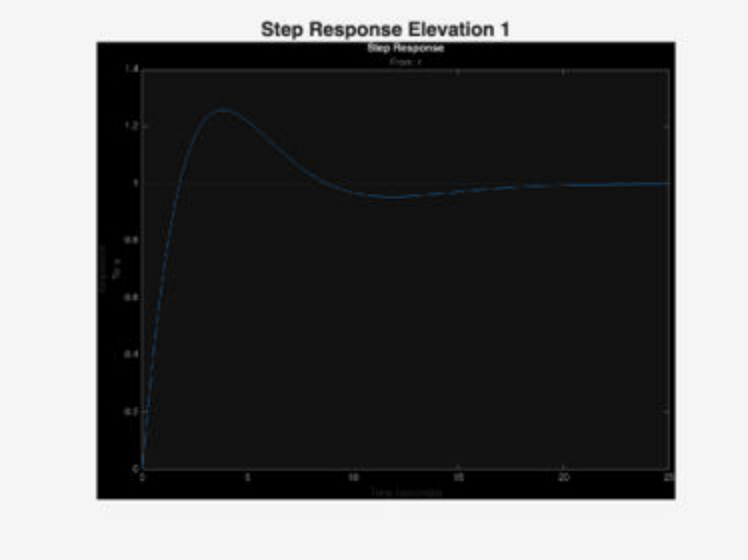
Question 1

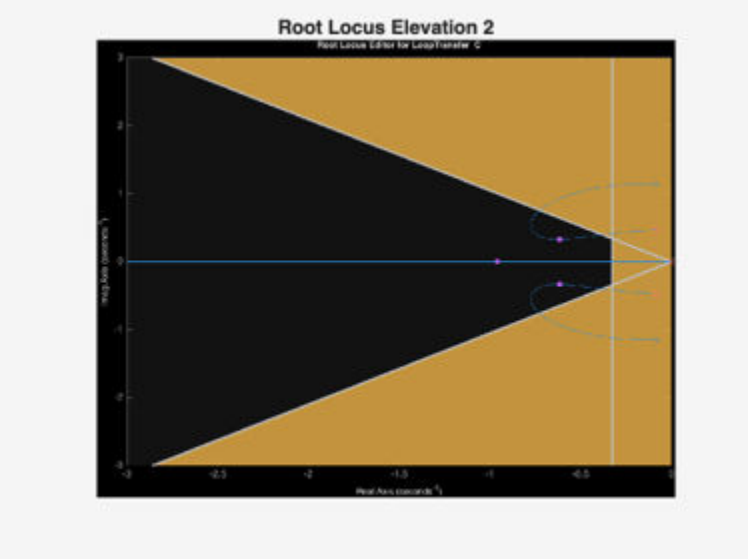
Root Locus and Step Response for Elevation 1

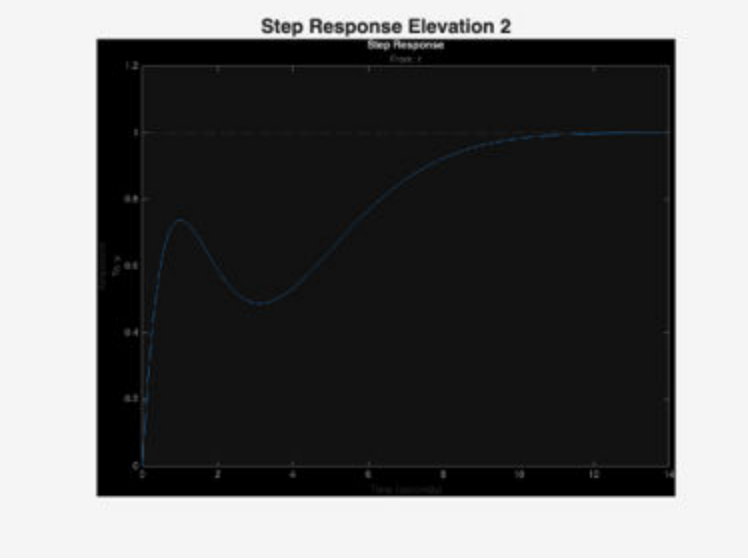
```
figure(1)
image(imread("Elev1RL.png"))
title('Root Locus Elevation 1')
axis off

figure(2)
image(imread("Elev1SR.png"))
title('Step Response Elevation 1')
axis off

% Root Locus and Step Response for Elevation 2
figure(3)
image(imread("Elev2RL.png"))
title('Root Locus Elevation 2')
axis off

figure(4)
image(imread("Elev2SR.png"))
title('Step Response Elevation 2')
axis off

% Root Locus and Step Response for Elevation 3
figure(5)
image(imread("Elev3RL.png"))
title('Root Locus Elevation 3')
axis off

figure(6)
image(imread("Elev3SR.png"))
title('Step Response Elevation 3')
axis off

% Root Locus and Step Response for Travel
figure(7)
image(imread("TravRL.png"))
title('Root Locus Travel')
axis off

figure(8)
image(imread("TravSR.png"))
```

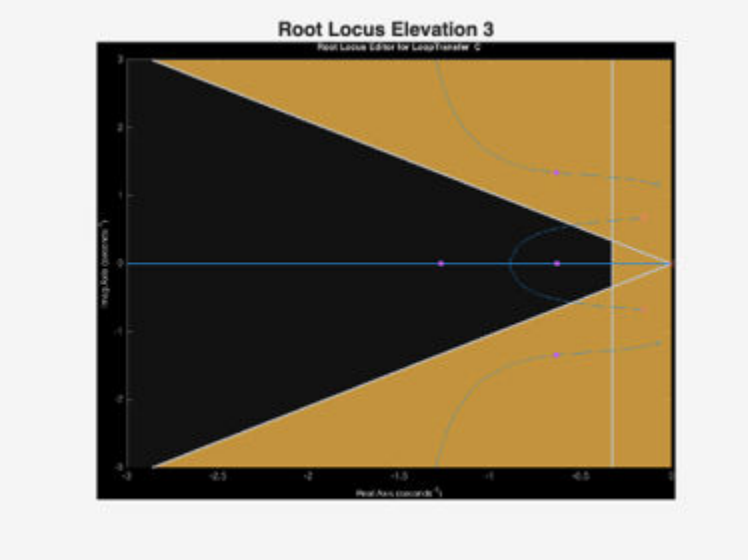
```
title('Step Response Travel')
axis off

% Analyzing the root locus and step response plots for each elevation:
%
% Elevation 1:
% - Root Locus: The root locus plot for Elevation 1 shows poles located in
%   the left half of the s-plane, indicating marginal stability. The poles
%   are close to the imaginary axis, contributing to an oscillatory response.
% - Step Response: The step response plot demonstrates significant overshoot,
%   with a settling time greater than 12 seconds. This suggests that the
system
%   is slow to stabilize, showing marginal stability with gradually decaying
oscillations.
%
% Elevation 2:
% - Root Locus: For Elevation 2, the root locus plot shows poles further left
%   in the s-plane compared to Elevation 1, resulting in a more stable system.
% - Step Response: The step response shows a reduced overshoot and faster
settling
%   time than Elevation 1, indicating a second-order system with improved
stability.
%
% Elevation 3:
% - Root Locus: The root locus plot for Elevation 3 shows poles significantly
%   farther left in the s-plane, indicating the most stable behavior among the
%   three elevations. The additional left-side pole provides further
stabilization.
% - Step Response: The step response plot for Elevation 3 shows minimal
overshoot
%   and a rapid settling time, satisfying stability criteria and indicating
that
%   the system stabilizes faster than the previous elevations.
```

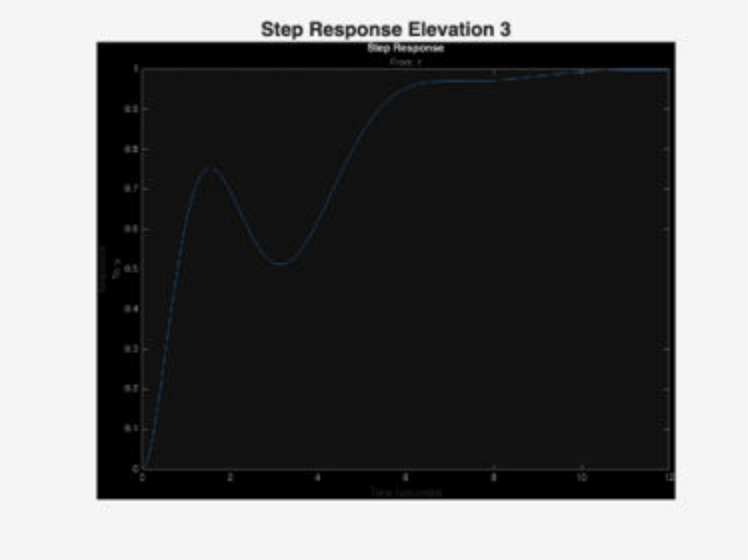


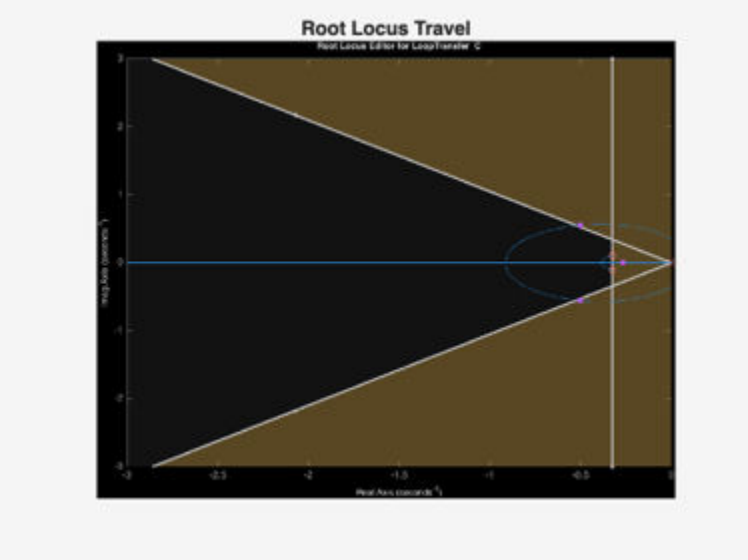

Step Response Elevation 1
Step Response

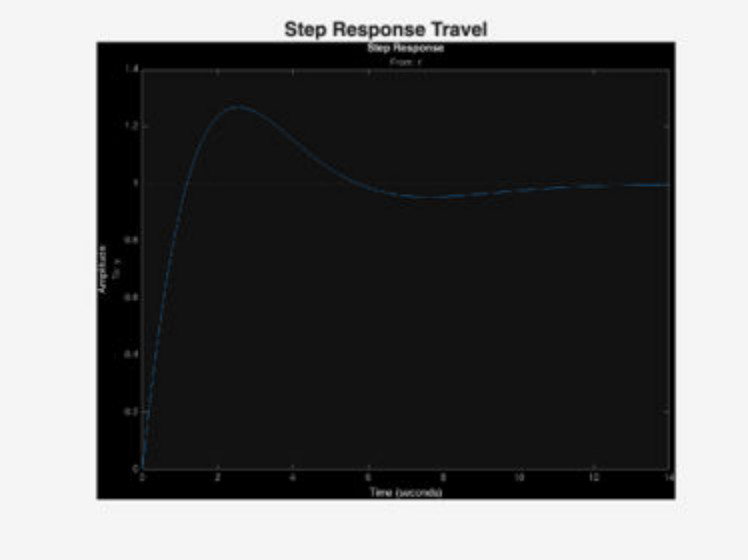

Root Locus Elevation 2
Root Locus Editor for LoopTransfer_C


Step Response Elevation 2
Step Response


Root Locus Elevation 3


Step Response Elevation 3
Step Response


Root Locus Travel



Question 2

Tranfer functions

```
G4_elev1 = tf(La*Kf, [Je, 0, 0])
G4_elev2 = tf(0.0294, [1.0000, 0.1538, 1.3288])
G4_elev3 = tf(0.0911, [1.0000, 3.1682, 1.7900, 4.0932])
G4_trav = tf(0.002362, [1 0 0])

% PID gain values of each tranfer function
PID1 = pid(C1)
PID2 = pid(C2)
PID3 = pid(C3)
PID_Trav = pid(C_Trav)
```

```
G4_elev1 =

   0.09157
  ---------
  1.044 s^2

Continuous-time transfer function.

G4_elev2 =

          0.0294
  ----------------------
  s^2 + 0.1538 s + 1.329

Continuous-time transfer function.

G4_elev3 =

               0.0911
```

```
  --------------------------------
  s^3 + 3.168 s^2 + 1.79 s + 4.093

Continuous-time transfer function.

G4_trav =

  0.002362
  --------
    s^2

Continuous-time transfer function.

PID1 =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 4.12, Ki = 0.539, Kd = 10.1

Name: C
Continuous-time PID controller in parallel form.

PID2 =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 11.6, Ki = 15.9, Kd = 69.3

Name: C
Continuous-time PID controller in parallel form.

PID3 =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 12.1, Ki = 19.3, Kd = 39.8

Name: C
Continuous-time PID controller in parallel form.

PID_Trav =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 351, Ki = 63.4, Kd = 540
```

```
Name: C
Continuous-time PID controller in parallel form.
```

Question 3

Comlete closed loop transfer function

```
feedback_elev1 = feedback(G4_elev1*PID1, 1)
feedback_elev2 = feedback(G4_elev2*PID2, 1)
feedback_elev3 = feedback(G4_elev3*PID3, 1)
feedback_trav = feedback(G4_trav*PID_Trav, 1)
```

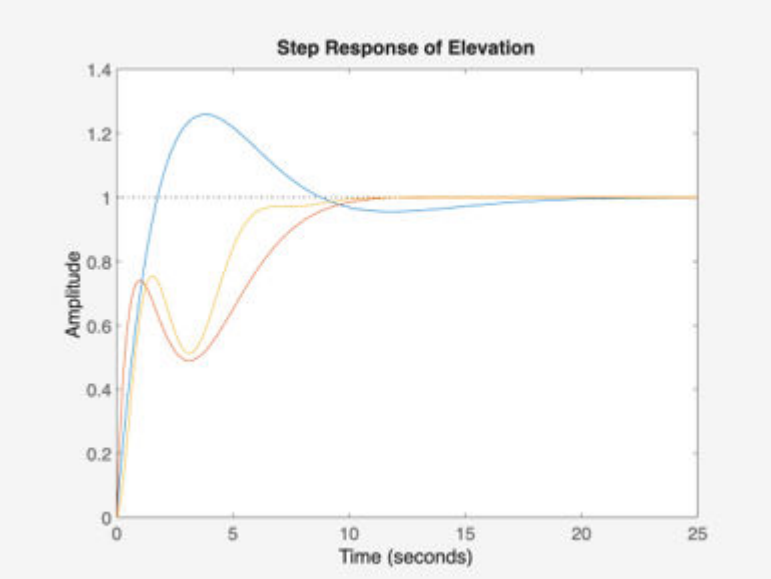
```
feedback_elev1 =

        0.9218 s^2 + 0.3769 s + 0.04935
  ----------------------------------------
  1.044 s^3 + 0.9218 s^2 + 0.3769 s + 0.04935

Continuous-time transfer function.

feedback_elev2 =

     2.036 s^2 + 0.34 s + 0.4683
  -------------------------------
  s^3 + 2.19 s^2 + 1.669 s + 0.4683

Continuous-time transfer function.

feedback_elev3 =

           3.623 s^2 + 1.106 s + 1.762
  ------------------------------------------
  s^4 + 3.168 s^3 + 5.413 s^2 + 5.199 s + 1.762

Continuous-time transfer function.

feedback_trav =

     1.274 s^2 + 0.8292 s + 0.1499
  ---------------------------------
  s^3 + 1.274 s^2 + 0.8292 s + 0.1499

Continuous-time transfer function.
```

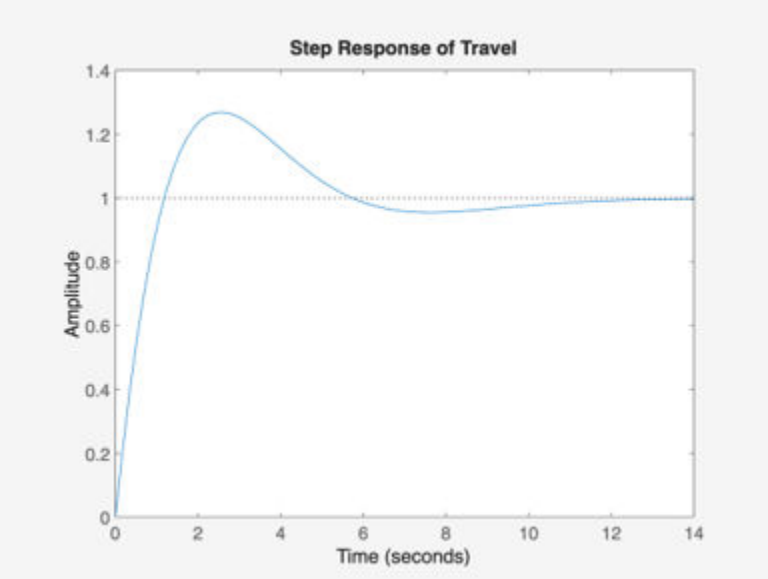
Question 4

Step response for elevation

```
figure(9)
step(feedback_elev1, feedback_elev2, feedback_elev3)
title('Step Response of Elevation')
```

```
% Step response of travel
figure(10)
step(feedback_trav)
title('Step Response of Travel')

% In summary, the Kd component of the PID controller is responsible for
% damping the system by reacting to the rate of change of the error signal.
% This is particularly important in the travel system, where rapid
% stabilization is necessary. The Kd gain is higher than both Kp and Ki
% because it helps reduce overshoot and oscillations, thus making the system
% more stable and less prone to fluctuations. The larger Kd value enhances
% the damping effect, allowing the travel system to settle more smoothly and
% avoid amplitude fluctuations. In contrast, Kp and Ki primarily adjust the
% system's responsiveness and steady-state behavior, but they don't provide
% the same level of stabilization as Kd. The increased Kd gain in the travel
% system step response reflects the need for strong damping to control the
% system’s behavior efficiently.
```





Question 5

Displaying characteristic information of the step response for different elevation systems

```
fprintf('Characteristic information of step response for elevation 1:\n')
stepinfo(feedback_elev1)

fprintf('Characteristic information of step response for elevation 2:\n')
stepinfo(feedback_elev2)

fprintf('Characteristic information of step response for elevation 3:\n')
stepinfo(feedback_elev3)

% Step response information for Elevation 1:
% Rise time is 1.3502, which is relatively quick, showing that the system
responds
% fairly quickly to a step input.
% Transient time and settling time are both 16.2627, indicating that the
system
% settles to its steady-state value at this time.
% The system exhibits an overshoot of 25.9069%, which suggests that it
initially
% exceeds the desired value before stabilizing.
% Peak occurs at 3.7764 seconds with a maximum value of 1.2591.

% Step response information for Elevation 2:
% Rise time is 7.5014, which is slower compared to elevation 1, indicating a
less
% responsive system.
% Transient and settling times are 9.7511, suggesting the system stabilizes
in this
% amount of time.
% There is a very small overshoot of 0.0822%, indicating minimal oscillations
before
```

```
% the system reaches its final value.
% The peak occurs at 14.0644 seconds, with a maximum value of 1.0008.
% Step response information for Elevation 3:
% Rise time is 5.1223, which is faster than elevation 2 but slower than
elevation 1.
% Transient and settling times are both 8.7398, indicating this system
stabilizes
% more quickly than elevation 2.
% The system has no overshoot (0%), showing that it reaches the steady-state
% value without exceeding it.
% Peak occurs at 13.7180 seconds, with a maximum value of 0.9992.

% In summary, the differences in rise time, overshoot, and settling time
across
% the different elevation systems can be attributed to the tuning of their
respective
% PID controllers, which control how the system reacts to input changes.
% Systems with faster rise times generally show higher overshoot, while those
with slower
% responses have lower overshoot but take longer to settle. The absence of
overshoot
% in elevation 3 shows a more dampened system response. This highlights how
adjusting
% PID controller gains can lead to different system behaviors, balancing
response speed
% with stability and overshoot.
```

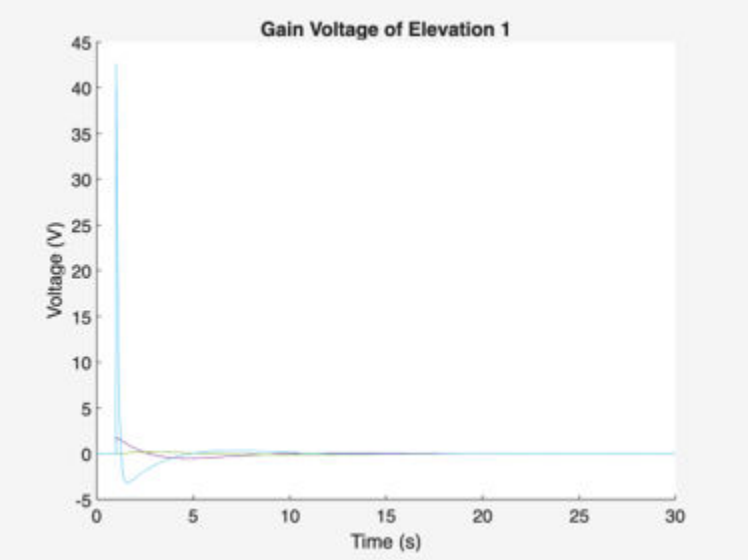
```
Characteristic information of step response for elevation 1:

ans =

  struct with fields:

         RiseTime: 1.3502
    TransientTime: 16.2627
     SettlingTime: 16.2627
      SettlingMin: 0.9552
      SettlingMax: 1.2591
        Overshoot: 25.9069
       Undershoot: 0
             Peak: 1.2591
         PeakTime: 3.7764

Characteristic information of step response for elevation 2:

ans =

  struct with fields:

         RiseTime: 7.5014
    TransientTime: 9.7511
     SettlingTime: 9.7511
      SettlingMin: 0.9001
      SettlingMax: 1.0008
```

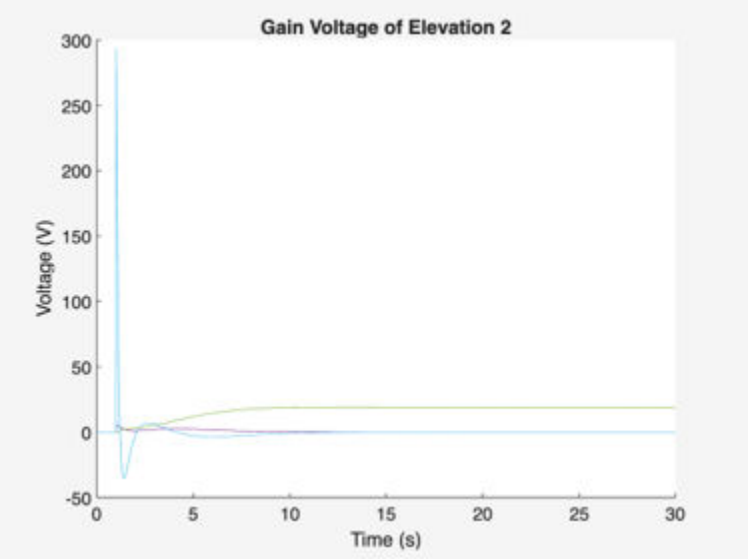
```
        Overshoot: 0.0822
       Undershoot: 0
             Peak: 1.0008
         PeakTime: 14.0644

Characteristic information of step response for elevation 3:

ans =

  struct with fields:

         RiseTime: 5.1223
    TransientTime: 8.7398
     SettlingTime: 8.7398
      SettlingMin: 0.9068
      SettlingMax: 0.9992
        Overshoot: 0
       Undershoot: 0
             Peak: 0.9992
         PeakTime: 13.7180
```

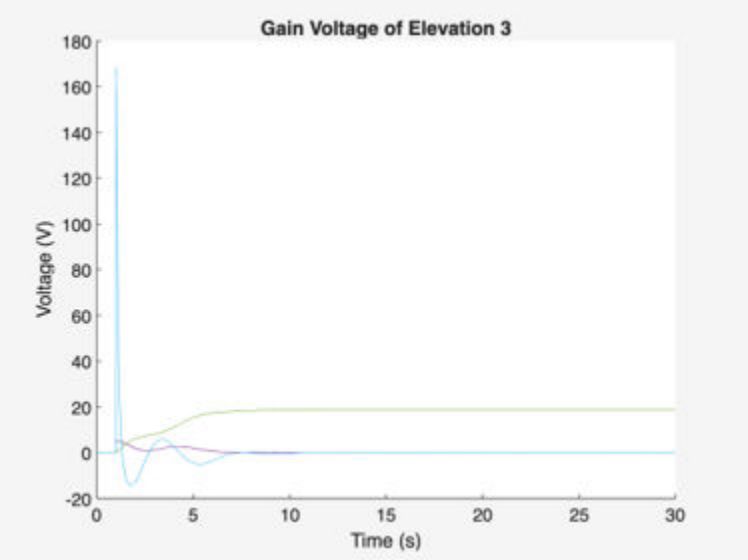
Question 6

```
%load elevationData1.mat
%load elevationData2.mat
%load elevationData3.mat

time = elev1(1, 1:end);

% Elevation 1
figure (11)
hold on
plot(CV1.time, CV1.signals(1).values)
plot(CV1.time, CV1.signals(2).values)
plot(CV1.time, CV1.signals(3).values)
hold off
title('Gain Voltage of Elevation 1')
xlabel('Time (s)')
ylabel('Voltage (V)')

% Elelvation 2
figure (12)
hold on
plot(CV2.time, CV2.signals(1).values)
plot(CV2.time, CV2.signals(2).values)
plot(CV2.time, CV2.signals(3).values)
hold off
title('Gain Voltage of Elevation 2')
xlabel('Time (s)')
ylabel('Voltage (V)')

% Elelvation 3
```

```
figure (13)
hold on
plot(CV3.time, CV3.signals(1).values)
plot(CV3.time, CV3.signals(2).values)
plot(CV3.time, CV3.signals(3).values)
hold off
title('Gain Voltage of Elevation 3')
xlabel('Time (s)')
ylabel('Voltage (V)')

% Discussion of Results
% By examining the gain voltages for each PID controller, we can observe
% how each individual gain (Kp, Ki, Kd) influences the control system
behavior.
% In the case of each elevation system, we notice that:
% - Kp (Proportional Gain) typically determines the system's initial response
to the error.
% - Ki (Integral Gain) helps eliminate steady-state error but may introduce
some oscillations
%   or slower stabilization as it accumulates error over time.
% - Kd (Derivative Gain), on the other hand, provides damping to the system,
as seen in
%   the elevation systems where the Kd gain is generally higher than Kp and
Ki.
%   This higher Kd value results in a more dampened response, reducing
overshoot
%   and improving the system's stability by slowing down the rate of change
of the error.

% In the plots of each elevation system's gain voltages, it can be observed
that:
% - The Kp gain contributes to the initial error reduction, causing the
system to respond
%   quickly but may leave some steady-state error.
% - The Ki gain acts to eliminate steady-state error, but as observed, it can
lead to
%   slower settling or small oscillations depending on its magnitude.
% - The Kd gain plays a critical role in damping, which is particularly
noticeable in
%   elevation systems where the Kd gain is set higher. This results in less
overshoot
%   and quicker stabilization of the system, ensuring better stability in the
face of
%   sudden changes in the input.

% In summary, the variation in Kp, Ki, and Kd across the three elevation
systems can be
% attributed to their specific tuning in the PID controller, which helps
optimize system
% performance, stability, and response time. Each PID controller is designed
to balance
% these factors, with the Kd term playing a significant role in reducing
oscillations and
% stabilizing the system, as evident from the plots.
```


Gain Voltage of Elevation 1
Voltage (V)
Time (s)
45
40
35
30
25
20
15
10
5
0
-5
0
5
10
15
20
25
30


Gain Voltage of Elevation 2
Voltage (V)
Time (s)
300
250
200
150
100
50
0
-50
0
5
10
15
20
25
30


Gain Voltage of Elevation 3
Voltage (V)
Time (s)
180
160
140
120
100
80
60
40
20
0
-20
0
5
10
15
20
25
30

Published with MATLAB® R2024a